IN THE SPECIFICATION:

Please see the attached amendments to certain pages of the specification.

Beginning on page 4 and continuing the paragraph through page 5, at line 4, change the paragraph to read as follows:

One related art method to which the method of the present invention generally relates is described in U.S. Patent No. 6,115,830 entitled "Failure Recovery For Process Relationships In A Single System Image Environment". This prior art method is a system for recovery of process relationships following node failure within a computer cluster. For relationship recovery, each node maintains set of care relationships. Each relationship is of the form the carer cares about the care target. Care relationships describe process relations such as parent-child or group leader-group member. Care relationships are stored at the origin node of their care targets. Following node failure, a surrogate origin node is selected. The surviving nodes then cooperate to rebuild vproc structures and care relationships for the processes that originated at the failed node at the surrogate origin node. The surviving nodes then determine which of their own care targets were terminated by the node failure. For each of the terminated care targets, notifications are sent to the appropriate carers. This allows surviving processes to correctly recover from severed process relationships.

At page 5, line 31, change the paragraph to read as follows:

Yet another related art method to which the method of the present invention generally relates is described in U.S. Patent No. 6,401,120 entitled "Method And System For Consistent Cluster Operational Data In A Server Cluster Using A Quorum Of Replicas". This prior art method is a method and system for increasing the availability of a server cluster while reducing its cost by requiring at a minimum only one node and a quorum replica set of storage devices (replica members) to form and continue operating as a cluster. A plurality of replica members maintain the cluster operational data and are independent from any given node. A cluster may be formed and continue to operate as long as one server node possesses a quorum (majority) of the replica members. This ensures that a new or surviving cluster has [[a]] at least one replica member that belonged to the immediately prior cluster and is thus correct with respect to the cluster operational data. Update sequence numbers and/or timestamps are used to determine the most updated replica member from among those in the quorum for reconciling the other replica members.

At page 18, line 11, change the paragraph to read as follows:

Fig. 1 illustrates a flowchart of the modeling of the Detection phase. The Detection Phase operates to determine what kind of clustering mechanism is deployed and thus what kind of operation can be used. This phase begins with a start bubble 10, followed by a process block to read the registry (block 11). An inquiry is followed, which checks to see whether or not the type <u>of</u> quorum is a local quorum (Diamond 12). If the answer to inquiry 12 is "YES", it is marked as local quorum (block 13), and proceeds to exit the detection phase (bubble 17). If the answer to inquiry 12 is "NO", another inquiry is made as to whether or not the type is a majority node set (Diamond 14). If the answer to inquiry 14 is "YES", it is marked as a majority node set (block 15). If the answer to inquiry 14 is "NO", it is marked as a shared disk quorum (block 16). The detection phase then proceeds to exit (bubble 17).

Beginning at the bottom of page 20 and continuing through the top of *page 21, at line 1, change the paragraph to read as follows:*

With reference to Fig. 2B, which follows from Fig. 2A, the continuation of the Revival phase is shown. This begins with an inquiry step 40 to check if the "active" node counts are one-half or less of the total node count. If the answer to inquiry 40 is "NO", the user is notified that the cluster is not revivable because it has already enough nodes to function correctly (block 41), which then ends (bubble 42). If the cluster is not working, but a majority of the nodes are active, then there is something else wrong. If the answer to inquiry 40 is "YES", another inquiry is followed (Diamond 43) to ask the user if they want to revive. If the answer to inquiry 43 is "NO", the user is notified that the cluster is not revived (block 44), and then ends out of the process (block 45).

At page 22, lines 20-22, change the paragraph to read as follows:

If the answer to inquiry 61 is "NO", the process quickly exits (bubble 62). If the answer to inquiry 61 is "YES", the cluster nodes are counted and looped through (block 63) to count the number of cluster nodes. At step 64, a query is asked -- is the total count =0? If the response is "NO" then step 65 will test the node response to where step 67 asks -- is the node available? If "YES" is the reply, then step 66 increments the active node count after which step 68 decrements the node count and returns to step 64. If “NO” is the reply at step 67, and the node is not available, the node count is decremented (block 68), and returns to check if the total count is equal to zero (block 64).

At page 22, line 26, change the paragraph to read as follows:

At step 64, if the query on total count =0? is "YES", then step 69 queries if the active node count is greater than one-half the total node count. This would indicate a majority of nodes. If the answer at step [[68]] 69 is "YES", this then moves to step 80 in Fig. 3B.